

PENGEMBANGAN EXTRACT METADATA DENGAN DATA SOURCING MASTER DATA MANAGEMENT PADA APLIKASI DATA GOVERNANCE MENGGUNAKAN OPEN SOURCE TOOLS

Ardy Rikardo*¹⁾, Tien Fabrianti Kusumasari²⁾, Ekky Novriza Alam³⁾

1. Telkom University, Indonesia
2. Telkom University, Indonesia
3. Telkom University, Indonesia

Article Info

Kata Kunci: Data Governance, Master Data Management, Metadata

Keywords: Data Governance, Master Data Management, Metadata

Article history:

Received 12 January 2023

Revised 19 January 2023

Accepted 14 February 2023

Available online 1 June 2023

DOI :

<https://doi.org/10.29100/jupi.v8i2.3491>

* Corresponding author.

Corresponding Author

E-mail address:

ardyrikardo@student.telkomuniversity.ac.id

ABSTRAK

Data merupakan sebuah aset yang penting bagi perusahaan atau organisasi. Data yang ada pada sebuah perusahaan atau organisasi pasti mengandung banyak informasi serta dengan jumlah data yang sangat banyak, maka untuk dapat menunjang kebutuhan tersebut perlu adanya Data Governance dalam pengolahan data. Saat ini, tidak sedikit dari perusahaan atau organisasi yang sangat mengandalkan kehadiran sebuah data untuk menjadi salah satu komponen utama yang dapat membuat sebuah keputusan yang sangat efektif serta efisien. Dengan data yang berjumlah sangat banyak per-lu adanya Master Data Management (MDM) untuk dapat menghasilkan sebuah kumpulan master data dengan struktur tabel yang sudah disesuaikan dengan business rules setiap perusahaan atau organisasi. Sedangkan, untuk dapat mengel-ola dan menyimpan data diperlukan sebuah database yang memiliki sebuah struktur tabel dan kolom yang disesuaikan dengan kebutuhan bisnis perusahaan atau organisasi, struktur tersebut dapat disebut sebagai business rules. Perlu adan-ya Metadata Management untuk mempermudah pembuatan tabel baru yang memiliki suatu pedoman dari tabel yang su-dah ada sebelumnya. Pada penelitian ini akan melakukan pengembangan extract Metadata dengan data sourcing MDM, serta menambahkan data sourcing eksternal agar memudahkan dalam proses import data pada MDM. Penelitian ini dil-akukan dengan menggunakan metode Iterative Incremental. Hasil pada penelitian ini merupakan extract Metadata dari data sourcing MDM, serta dapat import data sourcing eksternal untuk MDM.

ABSTRACT

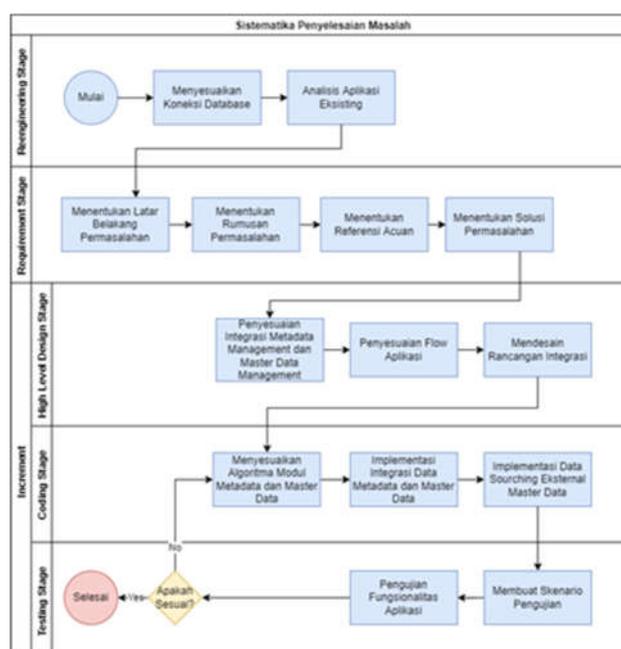
Data is an important asset for a company or organization. Existing data in a company or organization must contain a lot of information and with a very large amount of data, so to be able to support these needs it is necessary to have Data Governance in data processing. Currently, not a few companies or organizations rely heavily on the presence of data to be one of the main components that can make very effective and efficient decisions. With a large amount of data, it is neces-sary to have a Master Data Management (MDM) to be able to produce a master data collection with a table structure that has been adapted to the business rules of each company or organization. Meanwhile, to be able to manage and store data requires a database that has a table and column structure that is tai-lored to the business needs of the company or organi-zation, this struc-ture can be referred to as business rules. There is a need for Metadata Management to make it easier to create new tables that have a guideline from existing tables. In this study, we will develop a Metadata extract with MDM data sourcing, as well as add external data sourcing to facil-itate the process of importing data on MDM. This research was con-ducted using the Iterative Incremental method. The result of this research is to extract Metadata from MDM data sourcing, and can import external sourcing data for MDM.

I. PENDAHULUAN

Data merupakan sebuah aset yang penting bagi perusahaan atau organisasi. Data yang ada pada sebuah

perusahaan atau organisasi pasti mengandung banyak informasi serta dengan jumlah data yang sangat banyak. Saat ini, tidak sedikit dari perusahaan yang sangat mengandalkan kehadiran sebuah data untuk menjadi sebuah salah satu komponen utama dalam membuat sebuah keputusan yang sangat efektif serta efisien [1]. Namun data yang sangat banyak menjadi sebuah tantangan tersendiri untuk sebuah perusahaan dalam melakukan kontrol data. Data yang tidak terkontrol akan menyulitkan perusahaan dalam memelihara data dan menggunakan data. Untuk dapat mencegah hal tersebut perlu dilakukan suatu informasi dari data-data tersebut dengan menggunakan Metadata. Metadata berfungsi untuk perusahaan dapat mengetahui mengenai datanya, data yang merepresentasikan apa, berasal dari mana data tersebut, dan arti dari data tersebut [1]. Fungsi utama dari Metadata yaitu untuk menyediakan informasi dari suatu data yang ada untuk dijadikan sebuah guideline dalam pembuatan data baru untuk sebuah organisasi. Sedangkan data yang akan digunakan untuk ditinjau metadatanya adalah Master Data. Master Data merupakan data yang konsisten di seluruh sistem, data yang paling akurat tentang entitas bisnis yang penting [1]. Sedangkan untuk mendapatkan sebuah data yang baik dengan tingkat akurasi yang tinggi terdapat pada Master Data Management (MDM). MDM merupakan sebuah upaya untuk peningkatan kualitas dan kegunaan data perusahaan. Sebagian organisasi telah mengambil langkah untuk mengelola dan meningkatkan data dengan integrasi aplikasi dari masing-masing organisasi itu sendiri. MDM dianggap sebagai kumpulan praktik terbaik dengan fokus untuk menciptakan kumpulan Master Data yang digabungkan. MDM sangat berguna bagi sebuah organisasi [2]. Penelitian terkait Metadata ini merupakan lanjutan dari penelitian terdahulu mengenai implementasi *Metadata Repository*. Dalam penelitian terdahulu tersebut mengimplementasikan *Metadata Repository* pada aplikasi Data Governance. *Metadata Repository* digunakan untuk menyimpan informasi deskriptif. Informasi yang disimpan tersebut berupa Model Data yang digunakan untuk disimpan sesuai dengan kebutuhan organisasi dalam pembuatan Master Data, agar perusahaan dapat mengetahui terkait data yang dimiliki oleh organisasi [3]. Berdasarkan penelitian sebelumnya, terdapat keterbatasan bagi organisasi yang tidak memiliki sebuah Metadata melainkan hanya memiliki Master Data yang sudah di implemmentasikan pada sistem. Karena pada penelitian sebelumnya perusahaan harus memiliki sebuah Metadata terlebih dahulu yang akan disimpan pada Metadata Repository. Maka perlu adanya pengembangan pada modul MDM sehingga dapat melakukan penambahan *data sourcing* eksternal dengan format .csv dan .xls. Hal ini dapat mendukung tujuan dari aplikasi Data Governance yang dimana dapat di digunakan secara *open source*. Dengan adanya penambahan *data sourcing* eksternal, modul MDM dapat menerima *data sourcing* baru secara dinamis. Pada modul Metadata Management, metadata yang dapat di *extract* data yang berasal dari *local database* dan MDM. Hal ini dapat membantu sebuah organisasi dalam pelaksanaan integrasi atau sebagai *guideline* dalam pembuatan tabel baru Master Data yang disesuaikan dengan Master Data yang terdapat pada MDM dan setelah melakukan *extract* Metadata maka dapat mengunduh hasil *extract* Metadata dengan format .xls.

II. METODE PENELITIAN



Gambar 1 Sistematika Penyelesaian Masalah

Sistematika pada penelitian ini dilakukan untuk efektifitas serta fleksibilitas dari aplikasi eksisting Data Governance. Peneliti membandingkan beberapa metode yang menurut peneliti cocok diimplementasikan untuk penyelesaian masalah pada penelitian ini yaitu diantaranya metode Waterfall, Spiral dan Iterative Incremental. Metode ini membangun implementasi parsial dari sistem Total. Kemudian, secara perlahan menambahkan peningkatan fungsionalitas. Oleh karena itu, setiap rilis berikutnya akan menambahkan fungsi ke yang sebelumnya sampai semua fungsionalitas yang dirancang diimplementasikan [4]. Setelah melakukan perbandingan metodologi maka peneliti memilih metode iterative incremental. Metode ini memiliki beberapa fase dengan menyesuaikan topik penelitian untuk dapat mengembangkan *extract* metadata dengan *data sourcing* Master Data Management pada aplikasi *Data Governance* serta pada Master Data Management dapat menggunakan *data sourcing* eksternal pada gambar 1. Untuk memastikan bahwa penelitian sesuai dengan tujuan yang sudah di definisikan, maka untuk memastikan hal tersebut, perlu adanya pengujian terhadap hasil dari pengembangan penelitian ini menggunakan *Black Box Testing* yang akan dilakukan sesuai dengan skenario pengujian yang sudah dibuat oleh peneliti sesuai dengan fitur yang dikembangkan pada penelitian ini [5].

TABEL I OUTPUT FASE SISTEMATIKA PENYELESAIAN MASALAH

Tahapan	Teknik	Output
Reengineering	Code Review	Analisis aplikasi eksisting
Requirement	Requirements Development	Spesifikasi requirement aplikasi
High Level Design	Detailed Design	Usulan modifikasi flow aplikasi, data flow diagram, <i>wireframe</i>
Coding	Coding and Debugging	Hasil implementasi aplikasi
Testing	Fungsional Test	Hasil pengujian fungsional aplikasi

A. Data Governance



Gambar 2 Data Management

Data Governance didefinisikan sebagai pelaksanaan wewenang dan kontrol (perencanaan, pemantauan, dan penegakan) atas pengelolaan aset data. Fungsi Data Governance memandu semua fungsi pengelolaan data lainnya. Tujuan Data Governance adalah untuk memastikan bahwa data dikelola dengan baik, sesuai dengan kebijakan dan praktik terbaik [1].

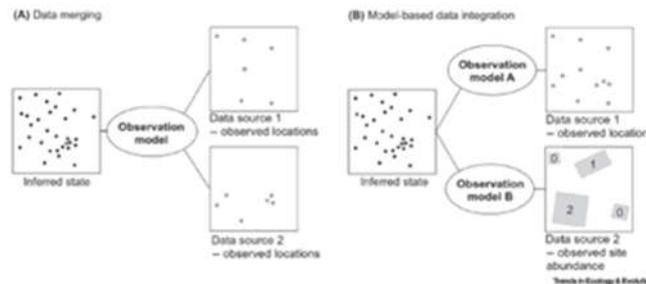
B. Metadata

Metadata bertujuan untuk menentukan informasi kontekstual kumpulan data untuk menggambarkan, menjelaskan, menemukan atau membuatnya lebih mudah untuk ditemukan, menentukan apakah suatu sumber daya relevan dan menggunakannya kembali dengan cara yang tepat. Metadata membantu dalam merepresentasikan informasi secara konsisten, merampingkan kemampuan alur kerja, dan melindungi informasi sensitif, terutama ketika kepatuhan terhadap peraturan diperlukan [1]. Metadata dapat dikumpulkan dari berbagai sumber. Apalagi Metadata dari aplikasi dan database sudah dikelola dengan baik, bisa langsung digunakan dan diintegrasikan. Namun, sebagian besar organisasi tidak mengelola Metadata dengan baik di tingkat aplikasi, karena Metadata sering dibuat sebagai produk sampingan dari pemrosesan aplikasi daripada sebagai produk akhir (yaitu, tidak dibuat dengan mempertimbangkan konsumsi). Seperti bentuk data lainnya, ada banyak pekerjaan dalam mempersiapkan Metadata sebelum dapat diintegrasikan [1].

C. Master Data Management (MDM)

Data master mengacu pada "set pengidentifikasi yang konsisten dan seragam dan atribut yang diperluas yang menggambarkan entitas inti dari suatu perusahaan, seperti pelanggan, calon klien, warga negara, pemasok, situs, hierarki, dan bagan akun". Manajemen data master dianggap sebagai "disiplin bisnis yang dipimpin oleh teknologi, yang dapat memungkinkan pemangku kepentingan bisnis dan TI dalam suatu perusahaan untuk bekerja sama dalam memastikan keseragaman, akurasi, penatagunaan, konsistensi semantik, dan akuntabilitas pejabat perusahaan, master bersama aset data", dan ini akan membantu pemangku kepentingan memperoleh keputusan yang lebih tepat dengan sumber data tepercaya, wawasan yang dapat ditindaklanjuti, dan pandangan holistik tentang pelanggan, produk, atau layanan mereka [6].

D. Data Integration



Gambar 3 Integrasi dan Penyatuan Data

Proses penyatuan data memiliki banyak label, misalnya data fusion, asimilasi, kombinasi, atau integrasi. Definisi mereka seringkali spesifik konteks, dan beberapa istilah memiliki beberapa arti. Daripada menguraikan semantik, kami hanya membedakan dua cara menyatukan data. Pengumpulan data mengasumsikan bahwa setiap perbedaan antara kumpulan data cukup kecil untuk diabaikan, atau diturunkan ke penyebut umum terendah misalnya, data kehadiran-ketidakhadiran dapat didegradasi menjadi data kehadiran-saja untuk digabungkan dengan data GBIF dalam SDM hanya-hadir [7].

E. Flowchart

Flowchart merupakan representasi diagram dari langkah-langkah suatu algoritma. Dalam flowchart, kotak dengan bentuk yang berbeda digunakan untuk menunjukkan jenis operasi yang berbeda juga. Kotak-kotak yang digunakan kemudian dihubungkan dengan garis berpanah yang dapat menunjukkan aliran atau arah yang harus dilalui untuk mengetahui langkah selanjutnya. Garis yang digunakan sebagai penghubung dikenal sebagai garis aliran [8]. Program flowchart merupakan tool yang sangat berguna untuk pengembangan program. Dikarenakan setiap kesalahan atau kelalaian dapat lebih mudah di temukan dari flowchart dibandingkan dengan dari program, karena flowchart adalah representasi bergambar dari logika sebuah program. Flowchart dapat diikuti dengan mudah dan cepat. Flowchart dapat berfungsi sebagai salah satu jenis dokumentasi, yang dapat membantu jika kebutuhan untuk memodifikasi program muncul di masa depan [8][9].

F. Data Flow Diagram (DFD)

Data Flow Diagram (DFD) merupakan sebuah artefak utama dan harus di buat untuk setiap sistem dalam pendekatan terstruktur. DFD adalah alat visual untuk menggambarkan model logika dan mengekspresikan transformasi data dalam suatu sistem. DFD mencakup mekanisme untuk memodelkan aliran data. Mendukung dekomposisi untuk menggambarkan rincian aliran data dan fungsi. DFD dapat menunjukkan aliran data dari luar ke dalam sistem dan menunjukkan bagaimana data dipindahkan dari satu proses ke proses lain [10]. DFD menggambarkan proses, penyimpanan data, dan entitas eksternal dalam bisnis atau sistem lain dan menghubungkan aliran data. Hanya ada empat simbol yang digunakan untuk DFD diantaranya *External Entity*, *Process*, *Data Flow*, *Data Store* [11].

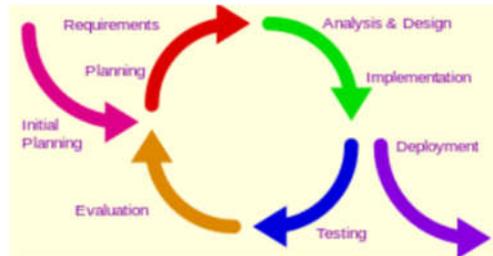
G. Open Source Tools

Pada pengembangan aplikasi manajemen data ini digunakan beberapa tools. Tools yang digunakan adalah tools yang bisa diakses secara terbuka atau biasa disebut *Open Source Tools*. *Open Source Tools* adalah alat yang memberikan akses gratis dan hak untuk mengubah source code untuk pengguna. Saat ini banyak tools dengan source code terbuka yang dapat dimanfaatkan oleh banyak pihak termasuk developer. Para developer biasanya akan membentuk sebuah komunitas yang memiliki kepentingan yang sama [12]. Komunitas tersebut yang menjadi akses atau acuan untuk Open Souce Tools. Terdapat beberapa aktivitas yang dilakukan pada komunitas tersebut seperti panduan instalasi, forum, dan dokumentasi [13]. Dengan banyaknya dokumentasi terkait Open Source Tools menjadi

alasan banyaknya pengguna yang memilih menggunakan Open Source Tools. Banyak tools yang dapat dimanfaatkan pengguna untuk mengembangkan aplikasi. Oleh karena itu, Open Source Tools digunakan pada penelitian ini. Adapun beberapa tools yang digunakan adalah Laravel, Pentaho, dan MySQL.

H. Metode Iterative Incremental

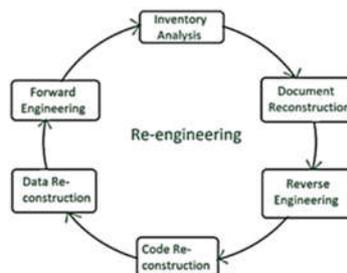
Metode Iterative Incremental merupakan sebuah metode yang menggabungkan elemen dari model Waterfall secara iteratif. Selain itu, setiap urutan linier akan menghasilkan sebuah peningkatan perangkat lunak yang dapat dikirimkan. Persyaratan dasar dibahas dalam kenaikan pertama, dan ini merupakan produk ini, namun banyak fitur tambahan yang tetap tidak terkirim pada kenaikan ini. Metode ini. Metode ini membangun implementasi parsial dari sistem total. Kemudian, secara perlahan menambahkan peningkatan fungsionalitas. Oleh karena itu, setiap rilis berikutnya akan menambahkan fungsi ke yang sebelumnya sampai semua fungsionalitas yang dirancang diimplementasikan [4].



Gambar 4 Metode Iterative Incremental [14]

I. Software Reengineering

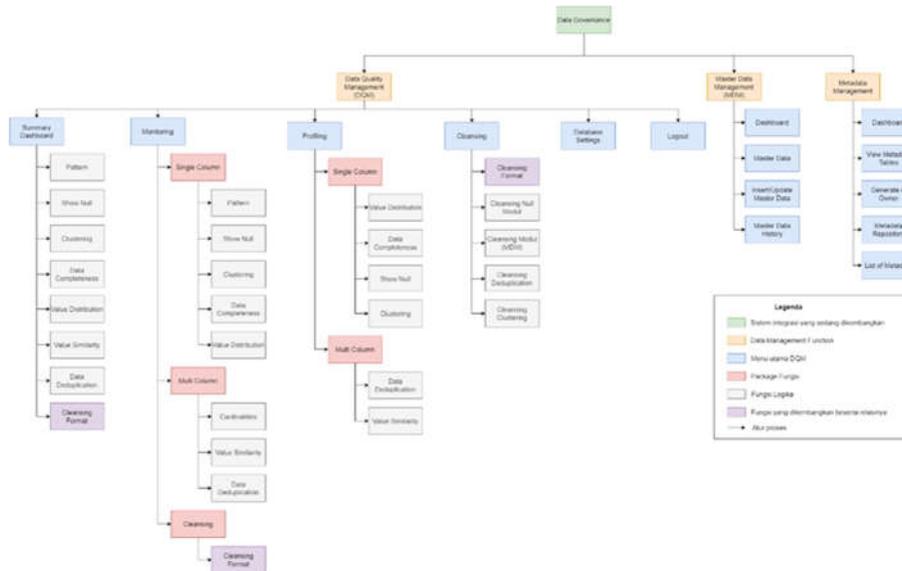
Pengembangan pada perangkat lunak biasanya dapat terjadi karena adanya pembaruan perangkat lunak, atau terdapat proses bisnis. Software Reengineering merupakan salah satu metode dimana proses perubahan di dasari dari berbagai latar belakang misalnya pemeliharaan yang lebih baik, peningkatan kinerja, perubahan arsitektur. Dengan menggunakan metode ini memiliki berbagai kelebihan dikarenakan dapat melakukan pengembangan atau perubahan pada perangkat lunak tanpa harus memulai dari awal, melainkan dapat mempertahankan perangkat lunak eksisting untuk dikembangkan lebih baik [15][16][17].



Gambar 5 Tahapan Software Reengineering

III. HASIL DAN PEMBAHASAN

Fase Reengineering merupakan sebuah tahap pertama dalam mengembangkan penelitian ini yang bertujuan untuk melakukan penyesuaian pada aplikasi eksisting agar dapat dijalankan, sehingga dapat memudahkan dalam mengidentifikasi fitur-fitur yang sudah di implementasikan. Dengan fase Reengineering memiliki manfaat ketika akan memberikan sebuah perubahan atau pengembangan yang lebih baik tanpa harus memulai dari awal kembali, melainkan melanjutkan implementasi yang sudah dilakukan Serta dapat mengetahui sebuah permasalahan yang terdapat pada aplikasi eksisting dengan meninjau fitur-fitur yang sudah diimplementasikan.



Gambar 6 Arsitektur Aplikasi Data Governance

Pada Gambar 6 merupakan arsitektur aplikasi *Data Governance*, baik yang telah dikembangkan maupun yang sedang dikembangkan. peneliti melakukan analisis terhadap modul Master Data Management dan Metadata Management. Master Data Management yaitu salah satu modul dari aplikasi Data Governance sebagai kumpulan dari Master Data. Sedangkan, Metadata Management yaitu salah satu modul dari aplikasi Data Governance sebagai panduan untuk master data. Metadata Management tidak dapat menggunakan data sourcing yang berasal dari Master Data Management sehingga berpengaruh ketika Metadata Management akan menggunakan data dari Master Data Management dan pada modul Master Data Management untuk saat ini hanya dapat digunakan pada studi kasus tertentu sehingga berpengaruh ketika akan digunakan untuk *general*. Kemudian pada penelitian ini akan berfokus untuk mengembangkan extract metadata dengan data sourcing Master Data Management dan menambahkan data sourcing eksternal pada modul Master Data Management.

Fase *Requirement* merupakan sebuah tahap kedua yang bertujuan untuk dapat mengetahui fitur-fitur eksisting yang sudah diimplementasikan pada aplikasi *Data Governance*. Adapaun pada penelitian ini berfokus pada modul Metadata dan Master Data Management.

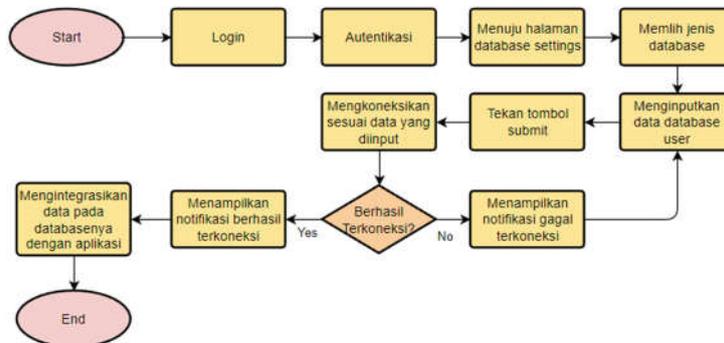
TABEL III ANALISIS MODUL EKSISTING

Fitur	Eksisting	Modifikasi	Penambahan
MDM-Dashboard	✓		
MDM-Master Data	✓		✓
MDM-Insert/Update Master Data	✓		✓
MDM-Master Data History	✓		
Metadata-Dashboard	✓		
Metadata-View Metadata Tables	✓	✓	
Metadata-Generate Id Owner	✓		
Metadata-Metadata Repository	✓		
Metadata-List of Metadata	✓		

Tabel II menjelaskan ada beberapa modul yang dilakukan modifikasi dan penambahan. Penelitian ini akan mengembangkan *extract* metadata yang akan melakukan sebuah penambahan pada fitur Master Data dan *Insert/Update* Master Data sehingga dapat menggunakan data sourcing eksternal di aplikasi *Data Governance*. Selain itu

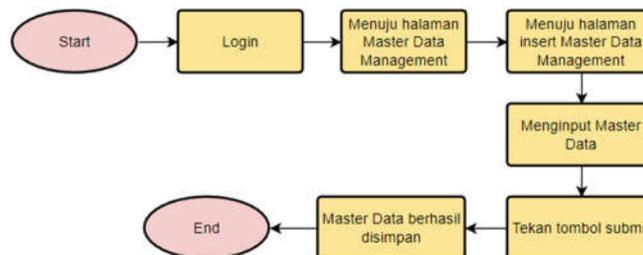
akan dilakukan modifikasi pada View Metadata Tables sehingga dapat menerima data dari Master Data Management maupun dari eksternal.

Fase *High Level Design* merupakan sebuah tahap ketiga yang bertujuan untuk melakukan perancangan modifikasi flow aplikasi, perancangan aliran data pada aplikasi yang dituangkan pada beberapa diagram yaitu *flowchart* diagram dan *data flow* diagram. Serta melakukan perancangan *user interface* untuk pengembangan yang menghasilkan sebuah *wireframe*.



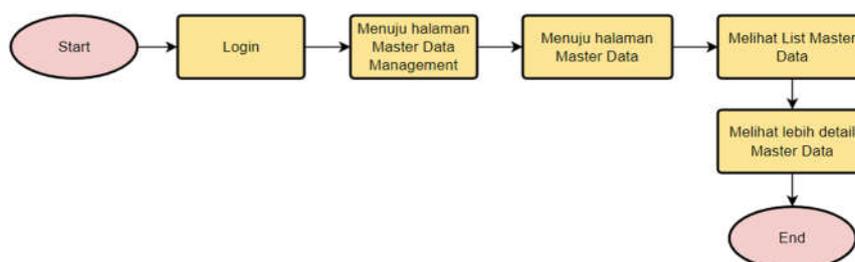
Gambar 7 Flowchart Input Database Settings

Flowchart pada Gambar 7 dimulai dengan user melakukan login pada aplikasi Data Governance terlebih dahulu, selanjutnya user menuju halaman database settings pada aplikasi Data Quality Management, kemudian memilih jenis database dan menginputkan data database agar dapat menggunakan database sesuai dengan yang dimiliki oleh user, selanjutnya aplikasi metadata akan melakukan koneksi dengan database yang sudah di inputkan oleh user, jika gagal akan menampilkan sebuah notifikasi gagal terkoneksi dan user dapat menginputkan kembali data database user, sedangkan apabila berhasil akan menampilkan sebuah notifikasi berhasil terkoneksi dan mengintegrasikan data pada database dengan aplikasi.



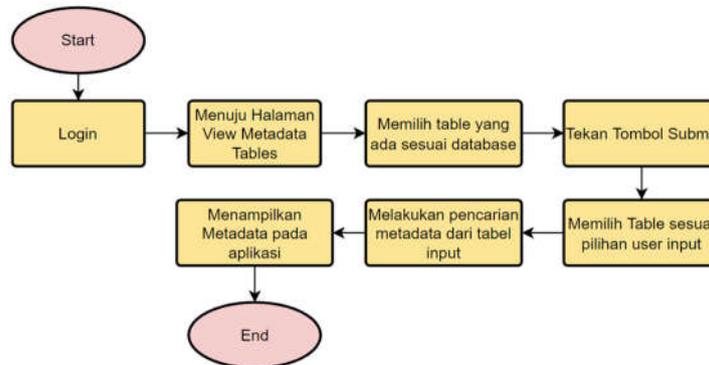
Gambar 8 Flowchart Insert Master Data Usulan

Flowchart pada Gambar 8 dimulai dengan user melakukan login pada aplikasi *Data Governance* terlebih dahulu, selanjutnya user akan menuju ke halaman Master Data Management kemudian user menuju ke halaman Insert Master Data Management untuk dapat menginputkan sebuah Master Data, user perlu menekan tombol submit apabila ingin data yang diinputkan tersimpan pada Master Data Management. Ketika user tekan tombol submit maka data akan tersimpan.



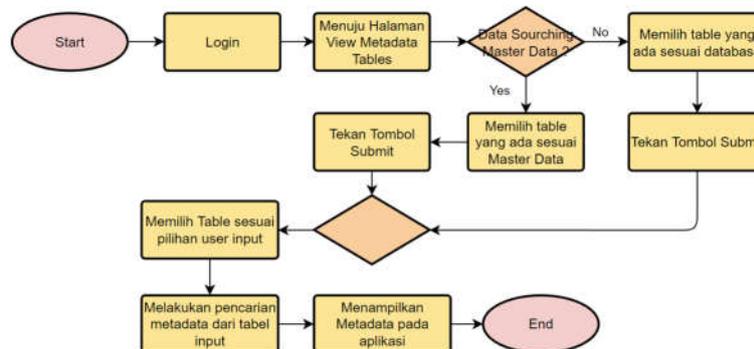
Gambar 9 Flowchart View Master Data Usulan

Flowchart pada Gambar 9 dimulai dengan user melakukan login terlebih dahulu pada aplikasi *Data Governance*, selanjutnya user akan menuju ke halaman Master Data Management lalu dilanjutkan dengan mengakses halaman Master Data, maka user dapat melihat semua Master Data yang tersimpan pada Master Data Management kemudian user dapat melihat detail master data dengan menekan tombol see details. Setelah menekan tombol see details maka user dapat melihat detail master data sesuai dengan nama *table* yang dipilih oleh user.



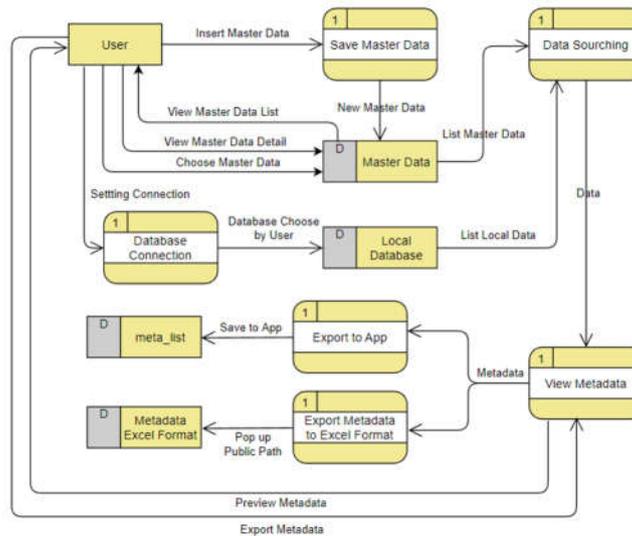
Gambar 10 Flowchart View Metadata Tables Eksisting

Flowchart pada Gambar 10 dimulai dengan user melakukan login terlebih dahulu pada aplikasi *Data Governance*, selanjutnya user akan menuju halaman View Metadata Tables yang akan di tampilkan pada user adalah sebuah section *dropdown*. Data yang akan di tampilkan adalah data yang berapada pada database user. User dapat memilih table yang di tampilkan pada section *dropdown* kemudian ketika user telah memilih *table* selanjutnya menekan tombol submit ketika *table* sudah dipilih oleh user, selanjutnya aplikasi akan menerima request *table* berdasarkan pilihan user input dan melakukan pencarian metadata dari *table* input. Ketika aplikasi sudah mendapatkan metadata dari *table* yang sudah di pilih oleh user, maka aplikasi akan menampilkan Metadata.



Gambar 11 Flowchart View Metadata Usulan

Flowchart pada Gambar 11 dimulai dengan user melakukan login terlebih dahulu pada aplikasi *Data Governance*, selanjutnya user akan menuju halaman View Metadata Tables. Pada Flowchart ini terdapat perbedaan dari sebelumnya dikarenakan pada View Metadata Eksisting tidak dapat menerima data dari Master Data Management. Maka dari itu untuk View Metadata Usulan dikondisikan untuk data menerima data dari Master Data Management. Ketika dilanjutkan kembali pada langkah di flowchart usulan maka terdapat pengondisian yang mengkondisikan ketika data sourcing yang dipilih adalah Master Data Management artinya data yang akan dipilih oleh user merupakan sebuah data Master Data. Sedangkan, ketika data sourcing yang di pilih selain Master Data Management maka akan menampilkan sebuah data yang terdapat pada database user. Ketika user sudah memilih data sourcing, selanjutnya menekan tombol submit agar aplikasi dapat menerima permintaan dari user. Aplikasi akan menerima permintaan dari user sesuai dengan *table* yang sudah dipilih, sehingga aplikasi dapat melakukan pencarian metadata dari *table* yang sudah dipilih dan akan menampilkan Metadata pada aplikasi.



Gambar 12 Data Flow Diagram

Pada Gambar 12 merupakan *Data Flow Diagram* (DFD) yang dapat menunjukkan aliran data dari luar ke dalam sistem dan menunjukkan bagaimana data dipindahkan dari satu proses ke proses lain. Proses aliran data diawali dengan user melakukan *insert* master data yang baru pada *Master Data Management* (MDM). Ketika sudah menentukan *data sourcing* yang akan di simpan pada MDM, selanjutnya aplikasi akan menyimpan *data sourcing* eksternal sesuai dengan yang di *import* oleh user. Ketika sudah berhasil menyimpan master data, maka master data yang baru saja disimpan akan tersimpan pada *database*. User dapat melihat list dari master data pada MDM dan juga dapat melihat lebih detail perihal data dari master data yang dipilih tersebut. Pada saat ingin melihat metadata dari sebuah master data yang berada pada MDM, maka ketika pada saat memilih *data sourcing*, apabila memilih *data sourcing* MDM maka aplikasi akan menampilkan data yang berupa list nama *table* yang berada pada MDM. Sedangkan, apabila memilih *data sourcing local* data maka aplikasi akan menampilkan data yang berupa list nama *table* yang berada pada *local database*. Namun untuk dapat memunculkan *local database* perlu melakukan *setting connection* terlebih dahulu. Selanjutnya ketika sudah memilih *data sourcing* maka data yang telah dipilih oleh user akan di proses untuk mendefinisikan metadata. Ketika telah selesai didefinisikan maka user dapat melihat *preview* dari metadata sesuai *table* yang sudah di pilih pada tahap sebelumnya. User dapat mengunduh hasil metadata dari master data yang dipilih sebelumnya dan dapat menyimpan hasil dari pendefinisian metadata ke *database*.

Fase *Coding* merupakan sebuah tahap keempat yang merupakan tahap ketika sudah menyelesaikan perancangan aplikasi. Setelah mengetahui rancangan aplikasi dapat melakukan proses pengembangan aplikasi hingga selesai sesuai dengan rancangan aplikasi.



Gambar 13 Halaman Insert Master Data .csv

Gambar 13 menampilkan halaman insert Master Data dengan ekstensi file .csv yang memiliki tujuan untuk melakukan penambahan *data sourcing* eksternal pada MDM dengan data yang berupa Master Data.



Gambar 14 Halaman Insert Master Data .xls

Gambar 14 menampilkan halaman insert Master Data dengan ekstensi file .xls yang memiliki tujuan untuk melakukan penambahan *data sourcing* eksternal pada MDM dengan data yang berupa Master Data.



Gambar 15 Halaman List of Master Data

Gambar 15 menampilkan halaman list of master data, ditampilkan seluruh Master Data yang telah berhasil di simpan pada Master Data Management.



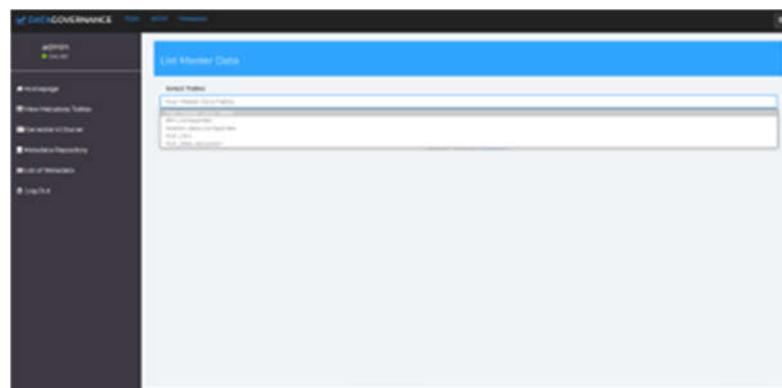
Gambar 16 Halaman Master Data Detail

Gambar 16 menampilkan halaman detail Master Data, ditampilkan detail data yang telah dipilih dari list Master data pada Master Data Management.



Gambar 17 Halaman Choose Data Sourcing

Gambar 17 menampilkan halaman untuk memilih *data sourcing*, terdapat dua *data sourcing* MDM dan *local database* yang dapat digunakan untuk *extract* metadata.



Gambar 18 Halaman Select Table From Master Data Management

Gambar 18 menampilkan halaman untuk menggunakan *data sourcing* dari data yang sudah tersimpan di MDM selanjutnya dilakukan *extract* metadata.



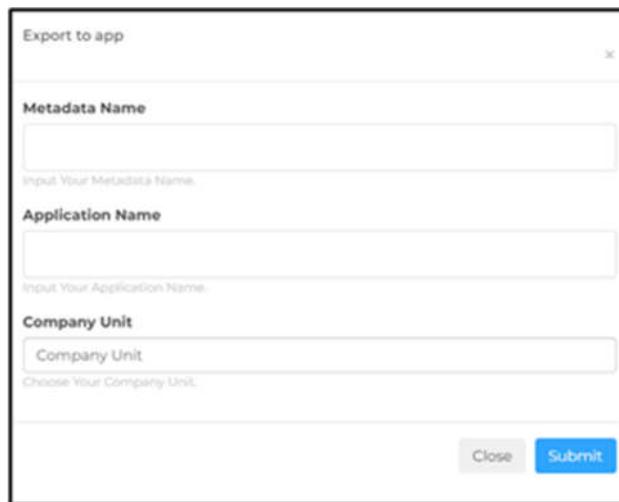
Gambar 19 Halaman Select Table From Database

Gambar 19 menampilkan halaman untuk menggunakan *data sourcing* dari *local database* dengan mengkonfigurasi pada *database settings* yang terdapat pada modul Data Quality Management (DQM).



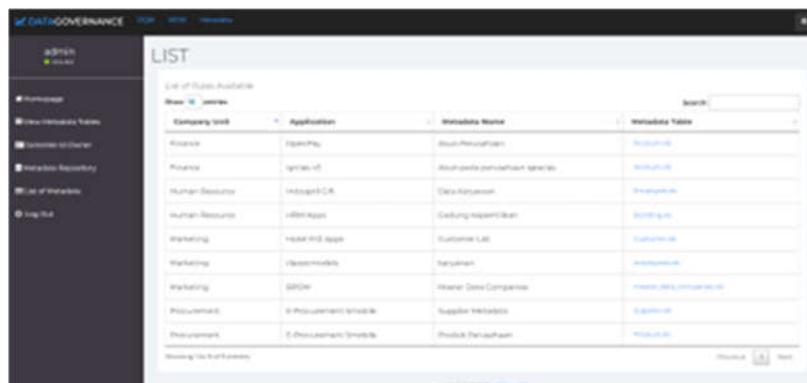
Gambar 20 Halaman View Database Metadata Tables

Gambar 20 menampilkan halaman sebuah informasi metadata dari tabel yang dipilih dari *data sourcing* MDM atau *local database*



Gambar 21 Popup Export Metadata to App

Gambar 21 menampilkan popup untuk menyimpan *extract* metadata ke metadata repository pada aplikasi *Data Governance*.



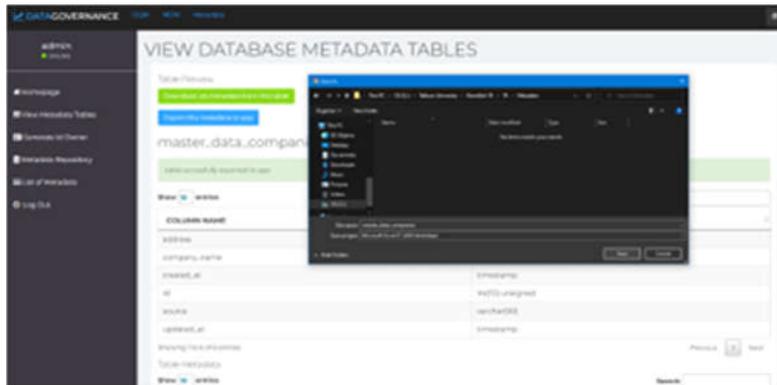
Gambar 22 Halaman List of Metadata

Gambar 22 menampilkan halaman yang berisi informasi metadata yang telah berhasil di *extract*, maka akan tersimpan pada halaman ini.



Gambar 23 Transformation Export to Excel

Gambar 23 merupakan Transformation dari Pentaho yang digunakan untuk melakukan proses *extract* metadata dengan *data sourcing* MDM atau *local database*. Hasil dari transformation ini adalah file excel dengan ekstensi *.xls*.



Gambar 24 Mengunduh hasil extract metadata

Gambar 24 merupakan tampilan ketika ingin mengunduh hasil *extract* metadata ke penyimpanan internal dengan format file yang disimpan berupa *.xls*.

Fase *Testing* merupakan fase penting dalam pengembangan aplikasi. Pada fase ini pengujian menggunakan *black box testing*. Pada fase ini menyiapkan skenario pengujian yang akan digunakan untuk pengujian dengan memastikan dari setiap fitur dapat berjalan dengan baik, terdapat sembilan skenario pengujian yang telah dibuat untuk mewakili dari setiap fungsionalitas sistem. Hasil dari pengujian yang sudah dilakukan, dari sembilan skenario ketika dilakukan pengujian telah berjalan sesuai dengan yang diharapkan. Maka, dapat disimpulkan pengembangan aplikasi pada penelitian ini berjalan dengan baik.

TABEL IIIII SKENARIO PENGUJIAN

No	Skenario	Prosedur	Hasil yang diharapkan	Hasil yang terjadi	Hasil sesuai? (Y/N)
1	User menambahkan <i>data sourcing</i> eksternal pada MDM	<i>User input data sourcing, User choose file, User Click Save</i>	User berhasil menambahkan <i>data sourcing</i> eksternal pada MDM	User berhasil menambahkan <i>data sourcing</i> eksternal dengan ekstensi .xls dan .csv pada MDM	Y
2	User melihat list master data	User mengakses tab Master Data pada modul MDM, User dapat melihat list Master Data	User dapat melihat list master data yang tersimpan pada MDM	User melihat list master data yang tersimpan	Y
3	User melihat detail data Master Data	User mengakses tab Master Data pada modul MDM, User dapat melihat list Master Data, User dapat melihat data Master Data	User dapat melihat data Master Data	User melihat data Master Data	Y
4	User melihat daftar table dari MDM	User memilih tab View Metadata Tables pada modul Metadata, User memilih <i>data sourcing</i> Master Data Management, User melihat data Master Data yang tersimpan pada list <i>dropdown</i>	User dapat melihat list Master Data dari MDM	User melihat list Master Data dari MDM	Y
5	User melihat daftar table dari <i>local database</i>	User memilih tab View Metadata Tables pada modul Metadata, User melihat data Master Data yang tersimpan pada list <i>dropdown</i>	User dapat melihat list data yang berasal dari <i>local database</i>	User melihat list data yang berasal dari <i>local database</i>	Y
6	User extract Metadata melalui aplikasi	User Memilih tab View Metadata Tables pada modul Metadata, User memilih <i>table</i> dari <i>data sourcing</i> Master data atau <i>local database</i> lalu <i>click submit</i> , Aplikasi akan menampilkan metadata dari <i>table</i> yang sudah dipilih	User dapat melihat hasil <i>extract</i> Metadata sesuai dengan <i>table</i> yang dipilih	User melihat hasil <i>extract</i> Metadata sesuai dengan <i>table</i> yang dipilih	Y
7	User export Metadata ke aplikasi	User <i>click export this Metadata to app</i> , ketika aplikasi menampilkan informasi Metadata dari <i>table</i> yang sudah di pilih, User <i>input</i> Metadata name, application name, dan company unit, <i>Click submit</i>	User dapat export Metadata ke aplikasi	User export Metadata ke aplikasi	Y
8	User download Metadata dengan ekstensi file .xls	User <i>click download</i> .xls ketika aplikasi menampilkan informasi Metadata dari <i>table</i> yang sudah di pilih, User memilih lokasi penyimpanan, <i>Click save</i>	User dapat export Metadata dengan ekstensi .xls	User export Metadata dengan ekstensi .xls	Y

9	User dapat melihat list Metadata	User mengakses tab list of metadata, Aplikasi menampilkan list Metadata	User dapat melihat list Metadata	User melihat list Metadata	Y
---	----------------------------------	-------------------------------------------------------------------------	----------------------------------	----------------------------	---

Hasil penelitian ini berfokus pengembangan pada modul MDM sehingga dapat melakukan penambahan *data sourcing* eksternal dengan format .csv dan .xls. Hal ini dapat mendukung tujuan dari aplikasi Data Governance yang dimana dapat di digunakan secara *open source*. Dengan adanya penambahan *data sourcing* eksternal, modul MDM dapat menerima *data sourcing* baru secara dinamis. Pada modul Metadata Management, metadata yang dapat di *extract* data yang berasal dari *local database* dan MDM. Hal ini dapat membantu sebuah organisasi dalam pelaksanaan integrasi atau sebagai *guideline* dalam pembuatan tabel baru Master Data yang disesuaikan dengan Master Data yang terdapat pada MDM dan setelah melakukan *extract* Metadata dapat mengunduh hasil *extract* Metadata dengan format .xls. Pada Gambar 25 merupakan standarisasi Metadata yang digunakan pada penelitian ini. Jika di dibandingkan dengan penelitian terdahulu yang berfokus pada penyimpanan Metadata dengan format .xls pada aplikasi Data Governance, tidak dapat melakukan *extract* Metadata dari sebuah Master Data [3].

COLUMN_NAME	COLUMN_TYPE	ORDINAL_POSITION	DATA_TYPE	COLUMN_KEY
NUMERIC_PRECISION	EXTRA	IS_NULLABLE	COLUMN_COMMENT	

Gambar 25 Standarisasi Metadata

IV. KESIMPULAN

Pengembangan *extract* metadata dengan data sourcing Master Data Management, dilakukan karena untuk menyesuaikan dengan real case yang terdapat pada perusahaan. Sehingga metadata menjadi sebuah perantara dalam membuat sebuah master data atau melakukan integrasi data. Sehingga metadata yang didapatkan dari master data eksisting dapat menjadi *guideline* untuk pembuatan master data. Penambahan data sourcing external pada modul Master Data Management diharapkan dapat memenuhi kebutuhan dari perusahaan, agar aplikasi Data Governance dapat digunakan oleh umum maka dengan adanya data sourcing external dapat digunakan dengan mudah dan tidak hanya terfokus pada studi kasus tertentu.

DAFTAR PUSTAKA

- [1] Dama International, *DAMA-DMBOK 2nd edition*. 2017.
- [2] M. Sarlin, "Mastering Product Data - Evaluation of Cross Domain Support in Master Data Management," 2021.
- [3] H. Nur Azizah, K. Tien Fabrianti, and A. Ekky Novrizza, "PENGEMBANGAN METADATA REPOSITORY MENGGUNAKAN OPEN SOURCE TOOLS," pp. 2-7.
- [4] A. Adel and B. Abdullah, "A Comparison Between Three SDLC Models Waterfall Model, Spiral Model, and Incremental/Iterative Model," *IJCSI Int. J. Comput. Sci. Issues*, vol. 12, no. 1, pp. 106-111, 2015, [Online]. Available: https://www.academia.edu/10793943/A_Comparison_Between_Three_SDLC_Models_Waterfall_Model_Spiral_Model_and_Incremental_Iterative_Model.
- [5] S. Nidhra, "Black Box and White Box Testing Techniques - A Literature Review," *Int. J. Embed. Syst. Appl.*, vol. 2, no. 2, pp. 29-50, 2012, doi: 10.5121/ijesa.2012.2204.
- [6] S. T. Ng, F. J. Xu, Y. Yang, and M. Lu, "A Master Data Management Solution to Unlock the Value of Big Infrastructure Data for Smart, Sustainable and Resilient City Planning," *Procedia Eng.*, vol. 196, no. June, pp. 939-947, 2017, doi: 10.1016/j.proeng.2017.08.034.
- [7] N. J. B. Isaac *et al.*, "Data Integration for Large-Scale Models of Species Distributions," *Trends Ecol. Evol.*, vol. 35, no. 1, pp. 56-67, 2020, doi: 10.1016/j.tree.2019.08.006.
- [8] A. B. Chaudhuri, *Flowchart and Algorithm Basics*. 2020.
- [9] A. M. Kuhn, *Code Complete*, vol. 47, no. 4. 2005.
- [10] Q. Li and Y.-L. Chen, "Data Flow Diagram," *Model. Anal. Enterp. Inf. Syst.*, pp. 85-97, 2009, doi: 10.1007/978-3-540-89556-5_4.
- [11] A. Aleryani and A. Y. Aleryani, "Comparative Study between Data Flow Diagram and Use Case Diagram Some of the authors of this publication are also working on these related projects: 1000 Researchers FCIT View project A MODEL TO MEASURE THE IMPACT OF CULTURE ON E-READINESS FOR E-GOVERNMENT," *Int. J. Sci. Res. Publ.*, vol. 6, no. 3, p. 124, 2016, [Online]. Available: www.ijsrp.org.
- [12] Y. Ye and K. Kishida, "Toward an understanding of the motivation of open source software developers," *Proc. - Int. Conf. Softw. Eng.*, vol. 6, pp. 419-429, 2003, doi: 10.1109/icse.2003.1201220.
- [13] S. Colford, "Explaining free and open source software," *Bull. Am. Soc. Inf. Sci. Technol.*, vol. 35, no. 2, pp. 10-14, 2009, doi: 10.1002/bult.2008.1720350205.
- [14] A. E. Chowdhury, A. Bhowmik, H. Hasan, and M. S. Rahim, "Analysis of the Veracities of Industry Used Software Development Life Cycle Methodologies," *AIUB J. Sci. Eng.*, vol. 16, no. 2, 2020, doi: 10.53799/ajse.v16i2.71.
- [15] A. Stojkov, L. Kazi, and M. Bla, "Software Reengineering with Object-Oriented n-Tier Architecture : Case of Desktop-to-Web Transformation," *Int. Conf. Inf. Technol. Dev. Educ. - ITRO 2020*, 2020.
- [16] M. Majthoub, M. H. Qutqut, and Y. Odeh, "Software Re-engineering : An Overview," pp. 266-270, 2018.
- [17] J. Hines, "Software engineering," *IEEE Spectr.*, vol. 35, no. 1, pp. 48-51, 1998, doi: 10.1109/6.645979.